Energy Minimization in Parallel Setting

Andrey Prokopenko
Jeremie Gaidamour, Jonathan Hu, Ray Tuminaro

16th Copper Mountain Conference on Multigrid Methods
21 March 2013

Sandia National Laboratories

Sandia National Laboratories is a multi-program laboratory operated by Sandia Corp, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

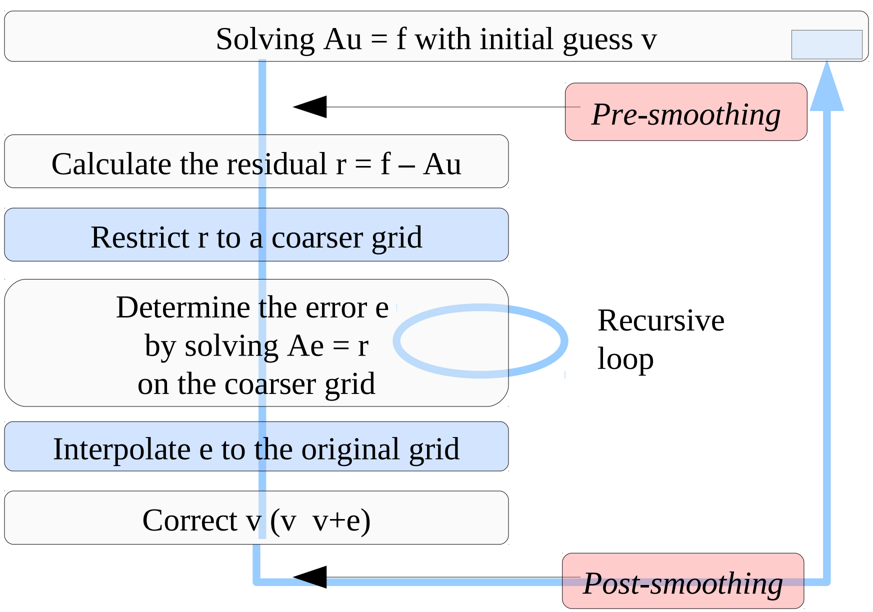
Sandia National Laboratories

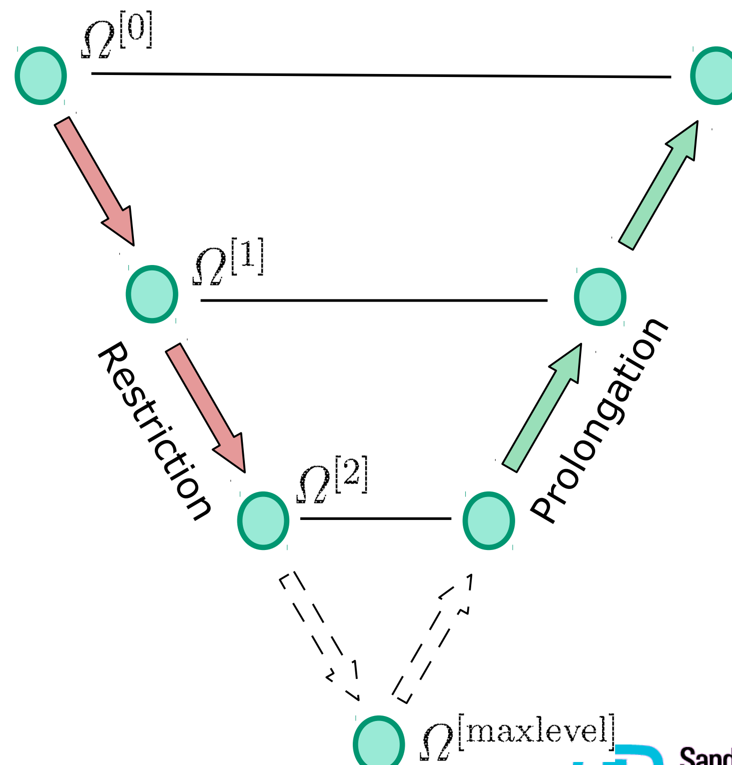
Outline

- Introduction
- Energy-minimization based AMG
  - Motivations
  - Algorithm
- Parallel implementation
- Numerical results
- Conclusion

AMG

- Iterative method for solving linear equations
- Commonly used as a preconditioner
- Idea: capture error at multiple resolutions using grid transfer operator:
  - **Smoothing** damps the oscillatory error (high energy)
  - **Coarse grid correction** reduces the smooth error (low energy)





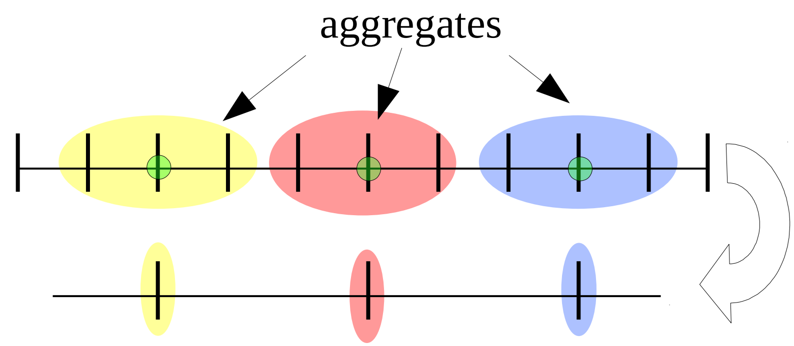
Prolongator requirements

Few desired properties

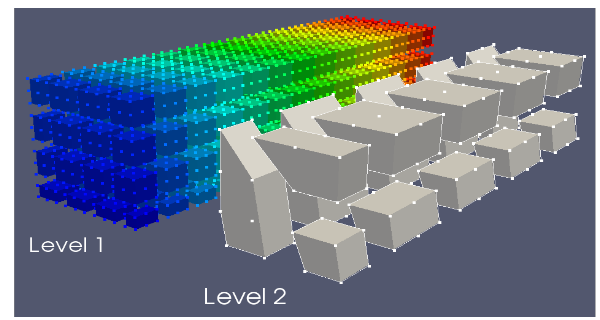
- **preservation of null space**: the span of basis functions on each coarse level should contain zero energy modes
- **minimization of energy**: basis functions on the coarse levels should have as small energy as possible
- **bounded intersection:** the supports of the basis functions on the coarse levels should overlap as little as possible.

Smoothed Aggregation

SA prolongator is constructed in a few steps



- Construct aggregates
  - Select a set of root nodes
  - Group unknowns into aggregates



- Construct tentative prolongator and coarse nullspace
  - Restrict fine nullspace onto aggregates
  - Do QR decomposition

    We satisfy $\mathrm{P}_{tent} B_c = B$

$$B = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ \vdots \end{bmatrix} \Rightarrow \begin{bmatrix} 1 & & \\ 1 & & \\ 1 & & \\ & 1 & \\ & 1 & \cdots \\ & 1 & \\ & & \ddots \end{bmatrix}$$

- Decrease energy of P_{tent} by smoothing

  $\mathrm{P} = (\mathrm{I} - \omega D^{-1} A) P_{tent}$

  May not satisfy $\mathrm{P}_{SA} B_c = B$

Energy minimization

Energy minimization

Energy minimization is a general framework.

Idea: construct the prolongator P by minimizing the energy of each column P_k while enforcing constraints.

Find P:

$$\mathrm{P} = \mathrm{argmin} \sum \|P_k\|_\chi$$

subject to

- specified sparsity pattern;
- nullspace preservation.

Advantages:

- Flexibility (input):
  - accept any sparsity pattern (arbitrary basis function support)
  - enforce constraints: important modes requiring accurate interpolation
  - choice of norm for minimization and search space
- Robustness

Constraint matrix

- Sparsity pattern
- B, B_c fine and coarse mode(s) requiring accurate interpolation

  Preservation of the nullspace: for instance $\mathrm{P1} = 1$

inputs

$$N = \begin{bmatrix} * & * \\ * & 0 \\ * & * \\ 0 & * \end{bmatrix} \qquad \mathrm{PB}_c = B \Leftrightarrow \begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \\ p_{31} & p_{32} \\ p_{41} & p_{42} \end{bmatrix} \begin{bmatrix} b^c_{11} \\ b^c_{21} \end{bmatrix} = \begin{bmatrix} b_{11} \\ b_{21} \\ b_{31} \\ b_{41} \end{bmatrix}$$

- Representation of the constraints in the algorithm:

$$\mathrm{XP} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ b^c_{11} & 0 & 0 & 0 & b^c_{21} & 0 & 0 & 0 \\ 0 & b^c_{11} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & b^c_{11} & 0 & 0 & 0 & b^c_{21} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & b^c_{21} \end{bmatrix} \begin{bmatrix} p_{11} \\ p_{21} \\ p_{31} \\ p_{41} \\ p_{12} \\ p_{22} \\ p_{32} \\ p_{42} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ b_{11} \\ b_{21} \\ b_{31} \\ b_{41} \end{bmatrix}$$

Constraint matrix

Two nullspace vectors:

$$P \begin{bmatrix} b^c_{11} & b^c_{12} \\ b^c_{21} & b^c_{22} \end{bmatrix} = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \\ b_{31} & b_{32} \\ b_{41} & b_{42} \end{bmatrix}$$

$$\begin{bmatrix} b^c_{11} & 0 & 0 & b^c_{21} & 0 & 0 \\ 0 & b^c_{11} & 0 & 0 & 0 & 0 \\ 0 & 0 & b^c_{11} & 0 & b^c_{21} & 0 \\ 0 & 0 & 0 & 0 & 0 & b^c_{21} \\ b^c_{12} & 0 & 0 & b^c_{22} & 0 & 0 \\ 0 & b^c_{12} & 0 & 0 & 0 & 0 \\ 0 & 0 & b^c_{12} & 0 & b^c_{22} & 0 \\ 0 & 0 & 0 & 0 & 0 & b^c_{22} \end{bmatrix} \begin{bmatrix} p_{11} \\ p_{21} \\ p_{31} \\ p_{12} \\ p_{32} \\ p_{42} \end{bmatrix} = \begin{bmatrix} b_{11} \\ b_{21} \\ b_{31} \\ b_{41} \\ b_{12} \\ b_{22} \\ b_{32} \\ b_{42} \end{bmatrix}$$

Energy-minimization algorithm

Find P:

$$\mathrm{P} = \operatorname{argmin} \sum \|P_k\|_\chi$$

subject to

- specified sparsity pattern;
- nullspace preservation.

Solve AP = 0
in a constrained Krylov space

- Definition of energy $\|\cdot\|_\chi$ depends on Krylov method
  - A for CG
  - $A^T A$ for GMRES

Energy minimization algorithm

Construct aggregates

$\mathcal{N} = |A||P^{(0)}|$ ▷ Select sparsity pattern

$D = diag(A)$ ▷ Diagonal preconditioner

$R = -AP^{(0)}$ ▷ Initial residual

$R = \textbf{enforce}(R, \mathcal{N})$ ▷ Enforce sparsity on R

$R = \textbf{project}(R, X)$ ▷ Enforce $RB_c = \mathbf{0}$

for i to iter **do**

    $Z = D^{-1}R$

    $\gamma =< R, Z >_F$

    if i is 1 **then**

        $Y = Z$

    else

        $\beta = \gamma/\gamma_{old};$

        $Y = Z + \beta Y$ ▷ New search direction

    end if

    $\gamma_{old} = \gamma$

    $Y_A = AY$

    $Y_A = \textbf{enforce}(Y_A, \mathcal{N})$ ▷ Enforce sparsity on Y_A

    $Y_A = \textbf{project}(Y_A, B_c)$ ▷ Enforce $Y_A B_c = \mathbf{0}$

    $\alpha = \gamma/ < Y, Y_A >_F$

    $P^{(i)} = P^{(i-1)} + \alpha Y$ ▷ Update prolongator

    $R = R - \alpha Y_A$ ▷ Update residual

A Special Case of Energy Minimization: SA

- Assume an initial guess P_0 satisfying $B = P_0 B_c$, i.e., it satisfies constraints of interpolating nullspace.
- Improve P_0 with one step of damped Jacobi:

$$P = (I - \omega D^{-1} A) P_0$$

- *P* still interpolates the nullspace. *P* can be rewritten as

$$P = P_0 - \omega D^{-1} A P_0 = P_0 + \Delta P$$

Note that $\Delta P B_c = 0$.

- SA can be viewed as one step of energy minimization with constraints specifying nullspace interpolation but not sparsity pattern enforcement.

Energy-minimization – Elasticity 3D

Lots of choices. We focus on 3 DOFs/nodes on the coarse grid

- 6 rigid body modes (3 translations & 3 rotations)
- CG to solve A P = 0 (effectively defines energy)
- P_0 & sparsity pattern are smoothed aggregation inspired
  - Initial Guess: tentative prolongator
  - Sparsity Pattern: $|S||P_{tent}|$, where S is either A, or filtered A
- Filtered matrix is defined using distance Laplacian + dropping for sparsity pattern
- A is still used to define energy (as opposed to filtered A)

Sandia National Laboratories

Comparison with Smoothed Aggregation

- SA: 6 DOFs/node
- Energy Minimization: 3 DOFs/node, 6 nullspace vectors

Tab. : Iteration count and *complexity* (lower complexity = faster run time) for increasing mesh sizes and stretch factors.

Mesh	$\epsilon = 1$				$\epsilon = 10$				$\epsilon = 100$			
	SA		Emin		SA		Emin		SA		Emin	
10^3	6	*1.30*	7	*1.07*	8	*2.81*	8	*1.22*	9	*3.21*	8	*1.24*
15^3	8	*1.19*	9	*1.05*	10	*2.32*	10	*1.15*	12	*2.54*	12	*1.16*
20^3	8	*1.24*	9	*1.06*	10	*2.59*	9	*1.18*	13	*3.05*	10	*1.20*
25^3	9	*1.26*	8	*1.07*	11	*2.76*	9	*1.20*	14	*3.04*	10	*1.20*
30^3	10	*1.22*	11	*1.05*	12	*2.52*	12	*1.17*	15	*3.06*	13	*1.19*
35^3	10	*1.24*	10	*1.06*	12	*2.66*	12	*1.18*	16	*3.03*	13	*1.19*
40^3	10	*1.26*	9	*1.06*	12	*2.77*	12	*1.19*	16	*3.21*	11	*1.21*

3.85x

complexity: $\dfrac{\sum_i nnz(A_i)}{nnz(A)}$

Parallel implementation

Energy minimization algorithm

Construct aggregates
$\mathcal{N} = |A||P^{(0)}|$ ▷ Select sparsity pattern

$D = diag(A)$ ▷ Diagonal preconditioner
$R = -AP^{(0)}$ ▷ Initial residual
$R = \mathbf{enforce}(R, \mathcal{N})$ ▷ Enforce sparsity on R
$R = \mathbf{project}(R, X)$ ▷ Enforce $RB_c = \mathbf{0}$

for i to iter **do**
  $Z = D^{-1}R$
  $\gamma =< R, Z >_F$
  if i is 1 **then**
    $Y = Z$
  else
    $\beta = \gamma/\gamma_{old};$
    $Y = Z + \beta Y$ ▷ New search direction
  end if
  $\gamma_{old} = \gamma$
  $Y_A = AY$
  $Y_A = \mathbf{enforce}(Y_A, \mathcal{N})$ ▷ Enforce sparsity on Y_A
  $Y_A = \mathbf{project}(Y_A, B_c)$ ▷ Enforce $Y_A B_c = \mathbf{0}$
  $\alpha = \gamma / < Y, Y_A >_F$
  $P^{(i)} = P^{(i-1)} + \alpha Y$ ▷ Update prolongator
  $R = R - \alpha Y_A$ ▷ Update residual

Energy minimization algorithm

Construct aggregates

$\mathcal{N} = |A||P^{(0)}|$ ▷ Select sparsity pattern

$D = diag(A)$ ▷ Diagonal preconditioner

$R = -AP^{(0)}$ ▷ Initial residual

$R = \textbf{enforce}(R, \mathcal{N})$ ▷ Enforce sparsity on R

$R = \textbf{project}(R, X)$ ▷ Enforce $RB_c = \mathbf{0}$

for i to iter **do**

  $Z = D^{-1}R$

  $\gamma =< R, Z >_F$

  if i is 1 **then**

    $Y = Z$

  else

    $\beta = \gamma / \gamma_{old};$

    $Y = Z + \beta Y$ ▷ New search direction

  end if

  $\gamma_{old} = \gamma$

  $Y_A = AY$

  $Y_A = \textbf{enforce}(Y_A, \mathcal{N})$ ▷ Enforce sparsity on Y_A

  $Y_A = \textbf{project}(Y_A, B_c)$ ▷ Enforce $Y_A B_c = \mathbf{0}$

  $\alpha = \gamma / < Y, Y_A >_F$

  $P^{(i)} = P^{(i-1)} + \alpha Y$ ▷ Update prolongator

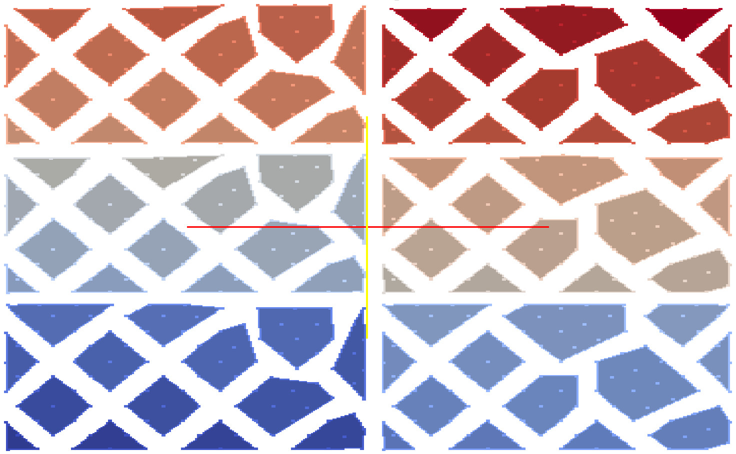
  $R = R - \alpha Y_A$ ▷ Update residual

Parallel aggregation

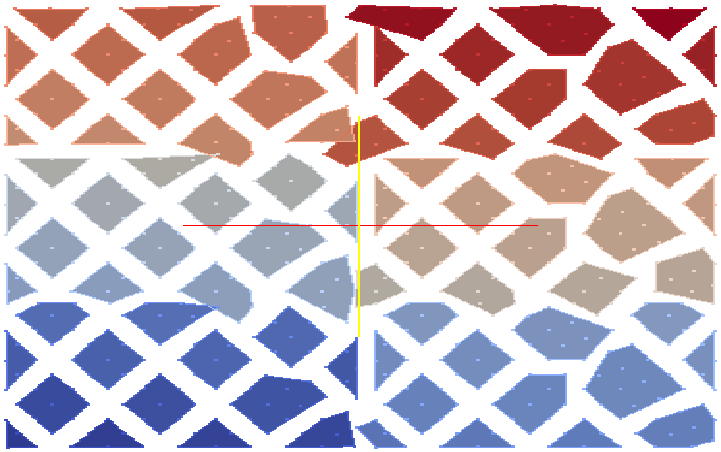
Two choices: coupled and uncoupled aggregation

- Uncoupled aggregation aggregates only inside a subdomain
- Coupled aggregation allows aggregates to cross subdomain boundary
- Coupled aggregation is more expensive, but has convergence similar to the serial case

Uncoupled



Coupled



Coupled aggregation

Couple aggregation algorithm:

1. Construct uncoupled aggregation in each subdomain (local procedure)
   - Some nodes are left unaggregated
2. Assign unaggregated vertices to adjacent root nodes from neighbor subdomains
   - Might require some arbitration
3. Create new root nodes and aggregates if we have multiple adjacent unaggregated nodes
4. Sweep remaining nodes into existing aggregates

Constraints in parallel

Let P have the following pattern and nullspace consist of two vectors

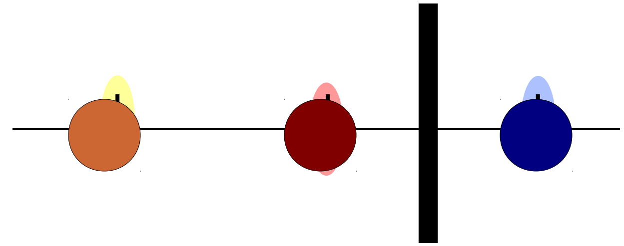
$$P \begin{bmatrix} b^c_{11} & b^c_{12} \\ b^c_{21} & b^c_{22} \end{bmatrix} = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \\ b_{31} & b_{32} \\ b_{41} & b_{42} \end{bmatrix} \qquad P = \begin{bmatrix} p_{11} & p_{12} \\ p_{21} & 0 \\ p_{31} & p_{32} \\ 0 & p_{41} \end{bmatrix}$$

$$\begin{bmatrix} b^c_{11} & 0 & 0 & b^c_{21} & 0 & 0 \\ 0 & b^c_{11} & 0 & 0 & 0 & 0 \\ 0 & 0 & b^c_{11} & 0 & b^c_{21} & 0 \\ 0 & 0 & 0 & 0 & 0 & b^c_{21} \\ b^c_{12} & 0 & 0 & b^c_{22} & 0 & 0 \\ 0 & b^c_{12} & 0 & 0 & 0 & 0 \\ 0 & 0 & b^c_{12} & 0 & b^c_{22} & 0 \\ 0 & 0 & 0 & 0 & 0 & b^c_{22} \end{bmatrix} \begin{bmatrix} p_{11} \\ p_{21} \\ p_{31} \\ p_{12} \\ p_{32} \\ p_{42} \end{bmatrix} = \begin{bmatrix} b_{11} \\ b_{21} \\ b_{31} \\ b_{41} \\ b_{12} \\ b_{22} \\ b_{32} \\ b_{42} \end{bmatrix}$$

$$\begin{bmatrix} b^c_{11} & b^c_{21} & 0 & 0 & 0 & 0 \\ b^c_{12} & b^c_{22} & 0 & 0 & 0 & 0 \\ 0 & 0 & b^c_{11} & 0 & 0 & 0 \\ 0 & 0 & b^c_{12} & 0 & 0 & 0 \\ 0 & 0 & 0 & b^c_{11} & b^c_{21} & 0 \\ 0 & 0 & 0 & b^c_{12} & b^c_{22} & 0 \\ 0 & 0 & 0 & 0 & 0 & b^c_{21} \\ 0 & 0 & 0 & 0 & 0 & b^c_{22} \end{bmatrix} \begin{bmatrix} p_{11} \\ p_{12} \\ p_{21} \\ p_{32} \\ p_{41} \\ p_{42} \end{bmatrix} = \begin{bmatrix} b_{11} \\ b_{12} \\ b_{21} \\ b_{22} \\ b_{31} \\ b_{32} \\ b_{41} \\ b_{42} \end{bmatrix}$$

Constraints in parallel

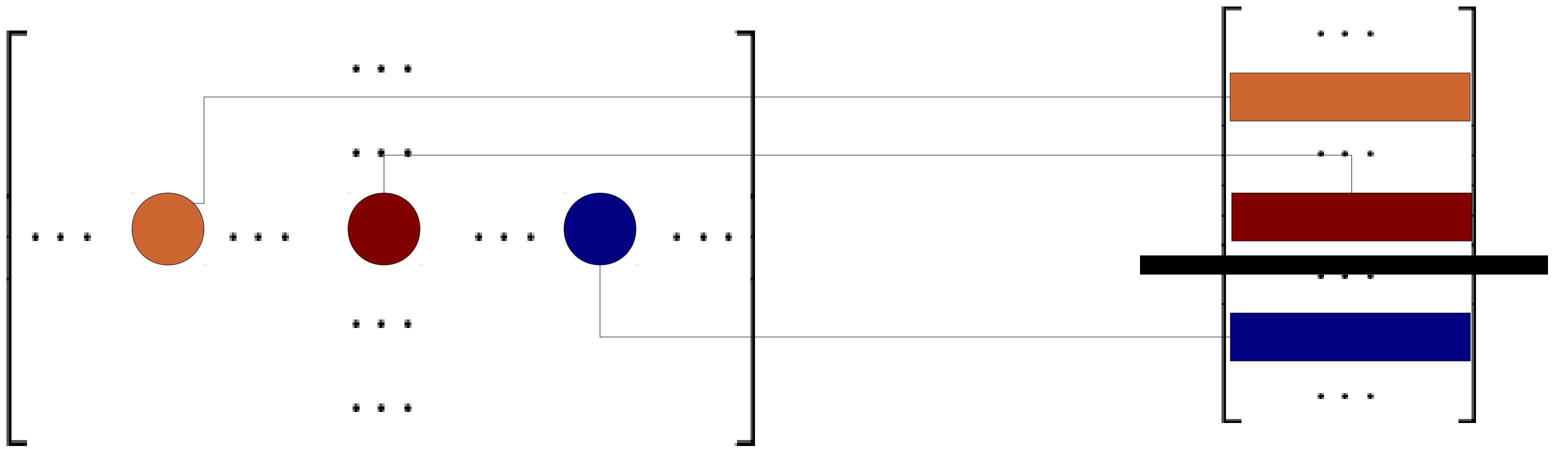
What does each block correspond to?



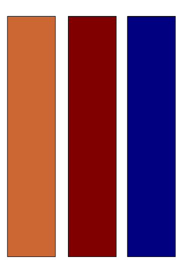
Consider a row of P with three nonzeros

Prolongator row

Coarse nullspace



Block of the constraint corresponding to the row



Energy minimization algorithm (updated)

Construct aggregates

$\mathcal{N} = |A||P^{(0)}|$ ▷ Select sparsity pattern

Import ghost components of nullspace vectors

$D = diag(A)$ ▷ Diagonal preconditioner

$R = -AP^{(0)}$ ▷ Initial residual

$R = \textbf{enforce}(R, \mathcal{N})$ ▷ Enforce sparsity on R

$R = \textbf{project}(R, X)$ ▷ Enforce $RB_c = \mathbf{0}$

for i to iter **do**

  $Z = D^{-1}R$

  $\gamma =< R, Z >_F$

  if i is 1 **then**

    $Y = Z$

  else

    $\beta = \gamma/\gamma_{old};$

    $Y = Z + \beta Y$ ▷ New search direction

  end if

  $\gamma_{old} = \gamma$

  $Y_A = AY$

  $Y_A = \textbf{enforce}(Y_A, \mathcal{N})$ ▷ Enforce sparsity on Y_A

  $Y_A = \textbf{project}(Y_A, B_c)$ ▷ Enforce $Y_A B_c = \mathbf{0}$

  $\alpha = \gamma / < Y, Y_A >_F$

  $P^{(i)} = P^{(i-1)} + \alpha Y$ ▷ Update prolongator

  $R = R - \alpha Y_A$ ▷ Update residual

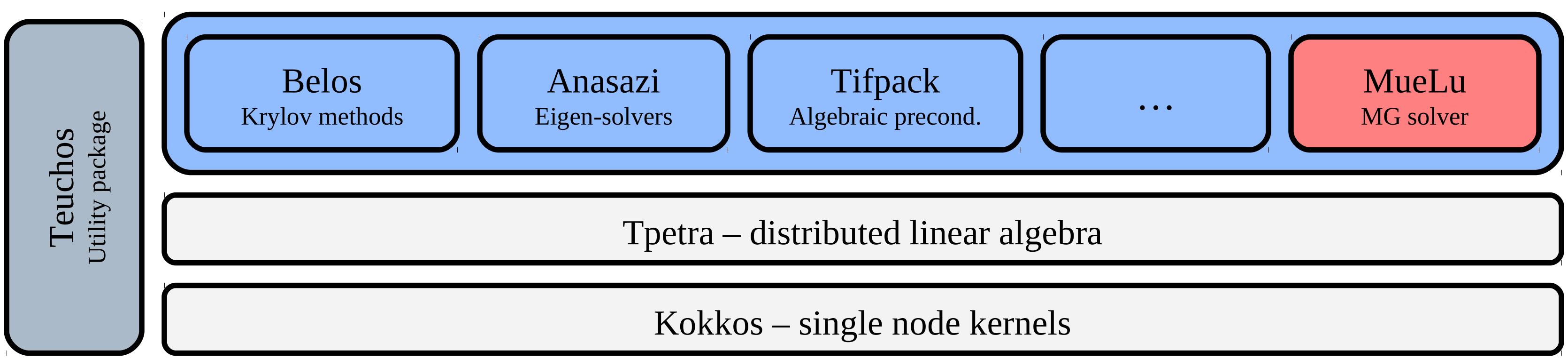
Sandia National Laboratories

MueLu

- Future package of the Trilinos project (to replace ML)
  - Massively parallel
  - Multicore and GPU aware
  - Templated types for mixed precision calculation (32-bit – 64-bit) and type complex

- Objective is to solve problem with billions of DOF on 100Ks of cores…

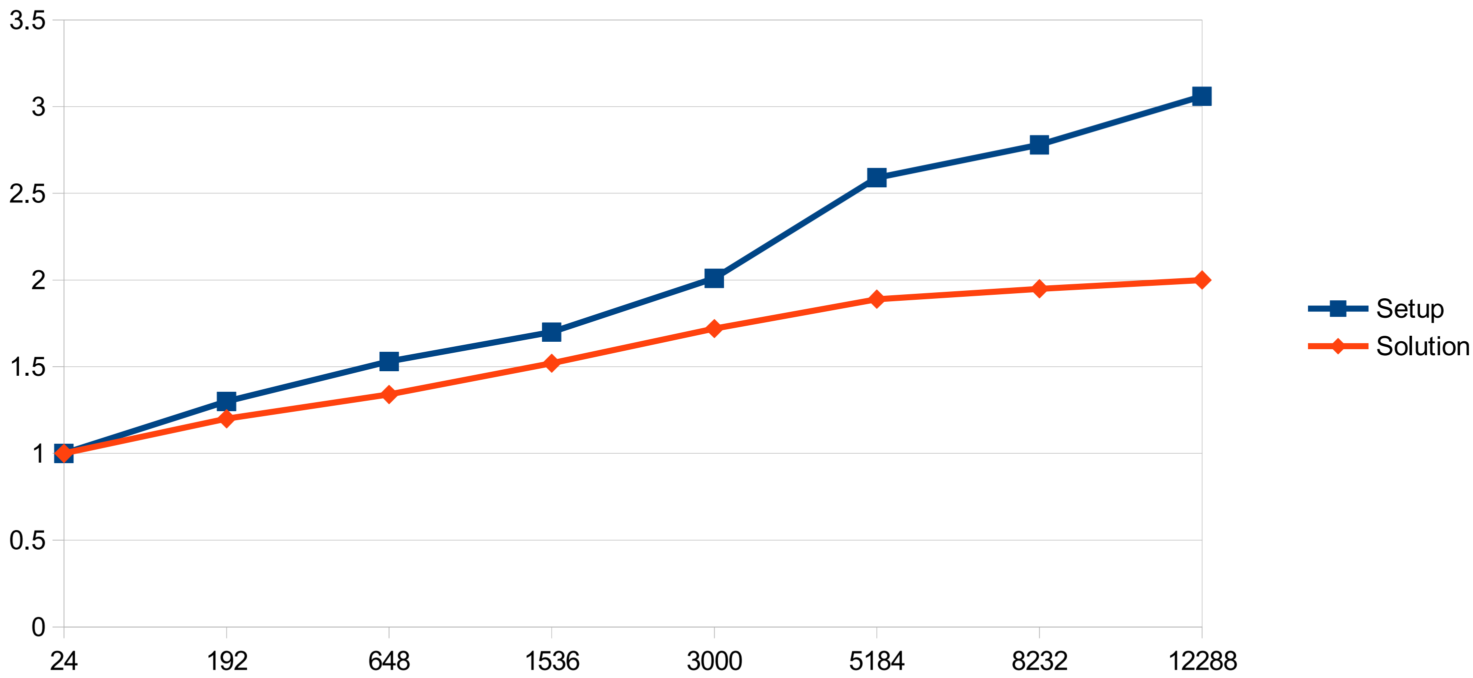
- Leverage the Trilinos software stack:



- Currently in development…

Numerical results – Laplace 3D

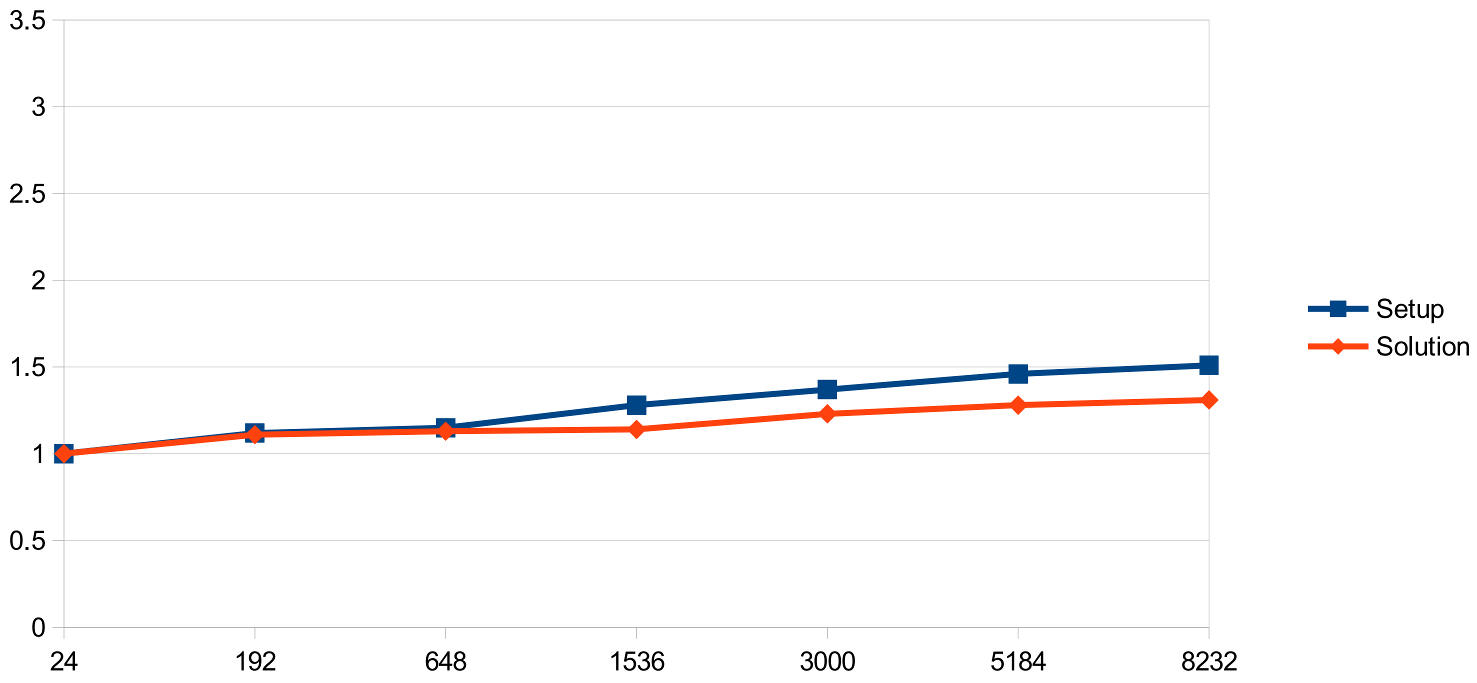
- Laplace 3D, 7 point stencil
- Energy minimization
  - 2 CG iterations
  - Initial guess: tentative prolongator
  - Sparsity pattern: same as SA



Sandia National Laboratories

Numerical results – Elasticity 3D

- Elasticity 3D, Poisson ratio 0.25
- Energy minimization
  - 2 CG iterations
  - Initial guess: tentative prolongator
  - Sparsity pattern: same as SA



Summary

- Energy minimization AMG is flexible
- Energy minimization AMG is suitable for parallelization
  - Standard parallel operations (MxM, BLAS1) are well known
  - Constraint application could be done locally storing ghost info
- Preliminary results show promise